William Thomas
P5: Identity Fraud from Enron Email

1. **Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those? [relevant rubric items: "data exploration", "outlier investigation"]**

   The objective of this report is to build an algorithm to identify Enron Employees who may have committed fraud (Persons of Interest) based on the Enron financial and email dataset. Machine learning focuses on automation, which means it can process data faster than people, who process data at a slow, manual, and iterative approach. Since the Enron financial and email dataset is quite large, using machine learning techniques towards this project will give me the opportunity to process the data at a faster and more efficient pace.

   Background on the data:

   Number of Employees: 146
   Number of POIs (Persons of interest): 18
   Number of non POIs: 128
   Number of Features: 21

   There are 14 financial features (Known as financial_features): Salary, deferral_payments, total_payments, loan_advances, bonus, restricted_stock_deferred, deferred_income, total_stock_value, expenses, exercised_stock_options

   There are 6 email features (Known as email_features): to_messages, email_address, from_poi_to_this_person, from_messages, from_this_person_to_poi, shared_receipt_with_poi

   Missing Values

   20 out of 21 features had missing values except for 'poi', Person of Interest.

   The variable poi is a Boolean (True = Yes, False = No), which identifies who is a person of interest and who isn't.

Email Features Missing Values:

Feature:	# of Missing Values:
‘to_messages’	60
‘email_address’	35
‘from messages’	60
‘from_this_person_to_poi’	60
‘from_poi_to_this_person’	60
‘shared_receipt_with_poi’	60

Financial Features Missing Values:

Feature	# of Missing Values:
‘restricted_stock’	36
‘deferral payments’	107
‘total_payments’	21
‘loan_advances’	142
‘director_fees’	97
‘restricted_stock_deferred’	128
‘total_stock_value’	20
‘shared_stock_options’	60
‘long_term_incentive'	80
'exercised_stock_options'	44
‘other’	53
‘deferred_income'	97
‘expenses’	51
‘salary’	51

Any Outliers?

Using the FindLaw Enron pdf file provided, outliers such as ‘LOCKHART, EUGENE E’ and ‘Travel Agency in the Park’ were found. LOCKHART did not have any financial data listed while ‘Travel Agency in the Park’ was listed as ‘Other’. Within the mini-project it was discovered that the row ‘Total’, which is in every financial feature, is also an outlier and will be excluded before moving on.

2. **What features did you end up using in your POI identifier, and what selection process did you use to pick them? Did you have to do any scaling? Why or why not? As part of the assignment, you should attempt to engineer your own feature that**

does not come ready-made in the dataset -- explain what feature you tried to make, and the rationale behind it. (You do not necessarily have to use it in the final analysis, only engineer and test it.) In your feature selection step, if you used an algorithm like a decision tree, please also give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values. [relevant rubric items: "create new features", "intelligently select features", "properly scale features"]

New Features

I decided to create 2 new features:

- fraction_from_poi Feature **Score**: 3.1280917481567374,
- fraction_to_poi Feature **Score**: 16.409712548035799

fraction_from_poi and **fraction_to_poi** by the total number of emails sent and received, respectively, might help us identify those have low amounts of email activity overall, but a high percentage of email activity with POIs. The higher the interaction a person has with a person of interest, the higher the probability that the person him or herself is a person of interest. Overall, **fraction_to_poi,** the new feature created, had the 5th highest feature out of all features which means that it's strong compared to the other features. **fraction_from_poi** had a low feature score and isn't considered to be strong compared to the other features.

Scaling:
Scaling has been done. MinMax scaling has been used since the units are different as well as the diversification of ranges among the features. "It shrinks the range such that the range is now between 0 and 1 (or -1 to 1 if there are negative values)."

SelectKBest

I decided to use SelectKBest, an automatic feature selection function. The SelectKBest contains a parameter known as 'k' which requires the optimal value to be assigned to it in order to output the best features. I decided to manipulate the 'k' parameter by changing to the number of features to determine which number had the best accuracy, precision and recall score using the Naïve Bayes Classifier.

1. GaussianNB(priors=None) **(K=5)**
   Accuracy: 0.85629 Precision: 0.49545 Recall: 0.32650 F1: 0.39361 F2: 0.35040

Total predictions: 14000 True positives: 653 False positives: 665 False negatives: 1347 True negatives: 11335

2. GaussianNB(priors=None) **(K= 6)**
   Accuracy: 0.86050 Precision: 0.51572 Recall: 0.38550 F1: 0.44120 F2: 0.40600
   Total predictions: 14000 True positives: 771 False positives: 724 False negatives: 1229 True negatives: 11276

3. GaussianNB(priors=None) **(K=7)**
   Accuracy: 0.85429 Precision: 0.48716 Recall: 0.37950 F1: 0.42664 F2: 0.39705
   Total predictions: 14000 True positives: 759 False positives: 799 False negatives: 1241 True negatives: 11201

4. GaussianNB(priors=None) **(K=8)**
   Accuracy: 0.85393 Precision: 0.48617 Recall: 0.39550 F1: 0.43617 F2: 0.41082
   Total predictions: 14000 True positives: 791 False positives: 836 False negatives: 1209 True negatives: 11164

K = 6 gave the maximum precision accuracy for classifiers while also giving me the 2nd best recall score. It was used to select the top 6 features with the highest scores.

- exercised_stock_options: 24.81
- total_stock_value: 24.18
- bonus: 20.79
- salary: 18.29
- fraction_to_poi: 16.41
- deferred_income: 11.458

3. What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms? [relevant rubric item: "pick an algorithm]

I ended up using three algorithms:

- Decision Tree Classifier
- GaussianNB Classifier

Tester_Classifier Results

- **Decision Tree Classifier**: Accuracy: 0.80757 Precision: 0.51572 Recall: 0.26900 F1: 0.28541 F2: 0.27533

- **GaussianNB:** Accuracy: 0.86050 Precision: 0.51572 Recall: 0.38550 F1: 0.44120 F2: 0.40600

GaussianNB did better compared to the decision tree classifier, but since GaussianNB has a limited amount of parameters for tuning.

4. **What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well? How did you tune the parameters of your particular algorithm? What parameters did you tune? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly explain how you would have done it for the model that was not your final choice or a different model that does utilize parameter tuning, e.g. a decision**

tree classifier). [relevant rubric items: "discuss parameter tuning", "tune the algorithm"]

Note: GaussianNB (My final classifier choice) did better compared to the decision tree classifier, but since GaussianNB has a limited amount of parameters for tuning I will use the DTC and explain what I would have done it it were my final choice.

Parameter tuning of an algorithm refers to the adjustment and optimization of a particular algorithm to improve it's fit on the test set. If done well, parameter tuning results in the best performance from an algorithm. However, if done wrong, it may affect the accuracy, precision and recall and make them poor.

I decided to tune the parameters for the Decision Tree Classifier:

clf_dt = DecisionTreeClassifier (criterion='gini', class_weight='balanced',

min_samples_split=3, min_weight_fraction_leaf=0.0,

presort=False, random_state=None, splitter='best')

5. **What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis? [relevant rubric items: "discuss validation", "validation strategy"**

Validation revolves around the process of using various techniques where a trained model is evaluated with a testing dataset. The testing data set is a separate portion of the same data set from which the training set is derived (70:30 ratio). A classic mistake you can make referring to your model is over-fitting. The model that has been over fit will perform well on the training set but score poorly on the test set. We can use cross validation techniques or reduce the number of features used in our dataset to avoid over fitting.

In order to validate my analysis, I made use of the train_test_split cross validation

6. Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something human-understandable about your algorithm's performance. [relevant rubric item: "usage of evaluation metrics"]

Precision score is the number of true positives divided by the total number of elements labeled as belonging to a positive class. Referring to the project, it is basically the ratio of number of true POIs correctly identified to the total number of records identified as Persons of Interests.

Recall score is defined as the number of true positives divided by the total number of elements that actually belong to the positive class. Specifically, it revolves around the total of true positives and false negatives, which are items which were not labeled as belonging to the positive class group but should have been. Referring to the project, it is the ratio of POIs correctly identified to the total amount of POIs in the dataset.

Following is the performance metrics for my Naïve Bayes Classifier:

accuracy: 0.86050

precision score: 0.51572

recall score: 0.38550